

The Comparison Of Dome And HMD Delivery Systems: A Case Study

Jian Chen^{*}, Deborah L. Hamm^{**}, R. Bowen Loftin^{***}, Laura C. Taylor^{****}, Ernst L. Leiss^{*}

^{*}University of Houston, ^{**}NASA/Johnson Space Center, ^{***}Old Dominion University, ^{****}Wyle Laboratories

Abstract

For effective astronaut training applications, choosing the right display devices to present images is crucial. In order to assess what devices are appropriate, it is important to design a successful virtual environment for a comparison study of the display devices. We present a comprehensive system, a VET, for the comparison of Dome and HMD systems on an SGI Onyx workstation. By writing *codelets*, we allow a variety of virtual scenarios and subjects' information to be loaded without programming or changing the code. This is part of an ongoing research project conducted by the NASA / JSC.

Keywords

Virtual environment testbed, projection-dome system, head-mounted display system, comparison, training

1. Background and Introduction

National Aeronautics and Space Administration (NASA) / Johnson Space Center (JSC) is seeking ways to deliver more effective training while lowering its cost. The use of Virtual Environment (VE) technologies has been proven to be an effective approach [1,2]. We have observed that using VE techniques in training applications has several advantages. First, VEs provide many hardware devices and software environments which serve as the simulators of work-related applications. The user has the feeling of "being there". Second, VEs allow for control of the interaction by the trainer and trainee, who experiences a "first-person" view [3]. It offers the possibility of providing innovative training

strategies. Finally, VEs enable training rehearsal which is especially useful to enhance learning. The simulators are quite forgiving in the way they tolerate mistakes. Besides that they are safe and cost-effective.

All VEs are "through the window" systems [4]. Visual feedback is without question the most dominant channel in the overall VE. Various display devices, such as full-immersive displays, spatial-immersive displays (SIDs), and virtual model displays (VMDs), have been designed to supply the user's eyes with either a stereoscopic or monoscopic view. However, previous studies have shown that no uniformly best display exists for all applications [5]. Instead, the suitability of a display is strongly related to the tasks to be performed in a virtual environment [6]. Therefore, it is important to be able to determine which device is appropriate for which application [7]. This paper presents methods for building an adaptive virtual environment system, called a Virtual Environment Testbed (VET), for the comparative evaluation of a Head-mounted display (HMD) and a projection-dome system (Dome) for pre-adapting astronauts to micro-gravity.

To make the testbed practical, several fundamental technological problems needed to be addressed: 1) choice of display generators and interfaces; 2) configuration of interface devices for interaction with applications; 3) interaction techniques; 4) software structure; and 5) acquisition of performance parameters.

The goal for the design of this VET was to address these five issues by applying as many VE design principles as possible. In addition, for efficiency and convenience, the system we developed should allow non-programmers to

^{*}{jchen2, coscel}@cs.uh.edu

^{**}dhamm@ems.jsc.nasa.gov

^{***}bloftin@odu.edu

^{****}ltaylor@ems.jsc.nasa.gov

design a new VE with little effort. We achieved our goals by using the smart object technique, delicate codelets, and the careful setup of a multi-model environment. In our system, we also provided tools to collect and visualize exposure data so that subject's performance can be evaluated effectively. The performance parameters recorded include task completion time, task accuracy, errors, and sickness status. Furthermore, a carefully designed pilot study is needed for obtaining a reasonable result. We address this issue in the future work.

There are three major motivations for this work. First, the dome display is a technology for constructing semi-immersive virtual environments capable of presenting high-resolution images. However, human factors issues related to it are not well understood. Second, neither qualitative studies nor rigorous evaluations of dome systems have been conducted. Third, a Virtual Research VR4® and a customized Dome delivery system are currently available at JSC for ground-based training tasks. Both systems can build a critical link between a virtual environment and the physical world. JSC personnel are eager to learn the merits of each for ground-based training applications in future work. A comparative study is interesting and relevant because the display devices are not equivalent.

The rest of the paper is organized as follows: We briefly survey related work in Section 2. Section 3 describes techniques for designing the VE Testbed. Three experiments from the pilot study are presented in Section 4. In Section 5, we present our conclusions. Finally, in Section 6, we describe some future directions.

2. Related Work

The potential user tasks involved in VE applications are enormous. A thoughtful approach to understanding the tasks is by splitting them into sub-tasks and analyzing the respective smaller tasks. In fact, interaction tasks can be classified as navigation, selection and manipulation, and

system control [8, 9]. With respect to the observer, all components can be egocentric or exocentric [10]. We have seen only two formalized evaluations that have been conducted on a "search" task. One was done by Bowman and coworkers [11], who compared a Virtual Research VR8® HMD with a CAVE® display. They built a virtual scenario of corridors. The corridors are textured polygons and no shadow is cast in the scene. Subjects were required to find several well-hidden targets. When designing the two VE systems for this scenario, different setups were used for the interaction mode, display mode, and main machine, as well as for the systems and libraries for presenting images. Finally, only one performance factor was considered for comparison: task completion time. This research provided guidelines for choosing an appropriate display for a search task. The authors concluded that the physical characteristics of the displays, user's experiences, and the method of doing pilot studies contributed significantly to subject performance. They also observed that the HMD was well suited for egocentric tasks.

Pausch and coauthors [12, 13] presented another way for comparing VE displays with a stationary monitor. The targets to be searched for are heavily camouflaged letters. In their system, the VR4® head-mounted display is used as a stationary display by fixing its position. The subject sits on a chair and holds a tracker in his hand. The two systems have the same resolution, field of view, image quality, and system setup. When executing pilot studies, error, level of fatigue, and search time were counted. The authors concluded that search performance decreased by roughly half when they changed from a stationary display to a HMD. In addition, the subject who wore an HMD reduced task completion time by 23% in later trials with the stationary display.

In conclusion, we see that two comparison methods have been applied. The first method for building a VE was based on an experiment where realistic systems were used and less

attention was paid to holding everything constant. In contrast, the second method for building an environment was based on an experiment where every factor was held constant in order to maintain the integrity of the statistics. So there exists a dilemma between getting good practical results and a simplified experimental design when designing a VE system. Which method we should employ depends on the applications and results [14]. Since ultimately our system is to be used for training applications, in the first stage of the design, we take the second method, that is, we try to maintain the two systems as similar as possible and minimize the factors that would affect the statistical results.

3. Application Design

3.1 Display Devices

In this study, a Virtual Research VR4® HMD and a customized dome display will be used for comparison. The HMD is a lightweight, rugged display with a 1.3" active matrix liquid color display (LCD). The resolution, field of view (FOV), and overlap are 640x480, 60°, and 100%, respectively.

The spherical dome is painted white and serves as a

projection surface. The inner surface is 3.7-meter in diameter. It is equipped with an Elumens projector [15] and a motion base (Figure 1). The resolution of this projector is 1280x1024 and the horizontal FOV is 180°. The motion base was not of interest for our study since we tried to minimize the differences between the HMD and the dome environments. The subject is seated upright inside the dome during the pilot study for both systems.

Obviously, the features of these two systems are quite different in numerous aspects. VR4® is portable and appropriate for some applications where the user works in isolation or needs to look around. However, the resolution is relatively poor. The subject may feel fatigue which is associated with prolonged use of HMD. Additionally, the interfaces they present to the user also vary. HMD provides a full-immersive environment but Dome is a type of spatial immersive display that presents a semi-immersive VE. Dome is better than HMD in balancing immersion and physical objects visualization and further offer a better sense of presence. Finally, the main drawback of Dome is its cost and the room space required to accommodate the system.

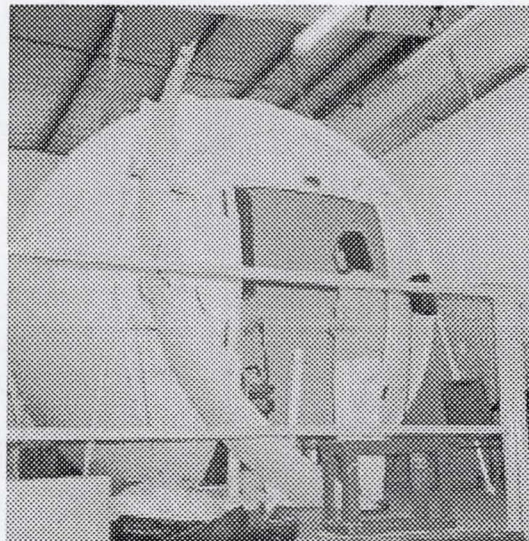


Figure 1 Projection-Dome Display

3.2 Tasks and Interaction Devices

Designing successful interaction depends on the task to be performed. The tasks used in our system are pick-and-release tasks. We made two assumptions here. One is that pick-and-release are the common tasks executed in training; the other is that they can be transferred effectively from a VE to the physical world.

When designing the VE, we considered the interface for our pick-and-release tasks as the combination of three common types of interactions. These include: body-centric navigation, hand-centric manipulation, and hand-centric selection. In our study, a joystick, a head-tracker (LogiTech™ ultrasonic), and a 3-D mouse (LogiTech™ ultrasonic) are integrated to support interactions.

3.3 Visual Databases

Our scenarios are similar to those described by Lampton and co-authors [16]. In our experiments, subjects are presented with a virtual room consisting of different colored and shaped objects. The tasks require subjects to move the objects on the left side of the room over to matching platforms on the right side of the room.

There are two levels, with different difficulties. Level one is made up of 10298 textured polygons. It represents a room about 21m long by 14m wide with a floor and four walls 1.6m high. On each side of the room are fifteen platforms. Each object is positioned on one of the platforms, and are in the shape of a torus, pyramid, cylinder, box, or sphere, combined with the color of green, red, or blue. Two obelisks in the middle of the room act as obstacles. Besides the basic geometric shapes, texture, shading, and shadows are drawn. These intrinsic physical properties can provide useful depth information and visual stimuli to the subject.

In level two, the room, objects, and platforms are the same as in level one. However, instead of using obelisks as obstacles, five different shapes of corridors are added. The paths through the corridors have almost the same number of left turns and right turns although the angles of the turns may be different (Figure 2). The model in this level contains 12434 textured polygons. In the pilot study, the subject may either travel without going through a prespecified path or must go through a path that varies with the type of the object picked up.

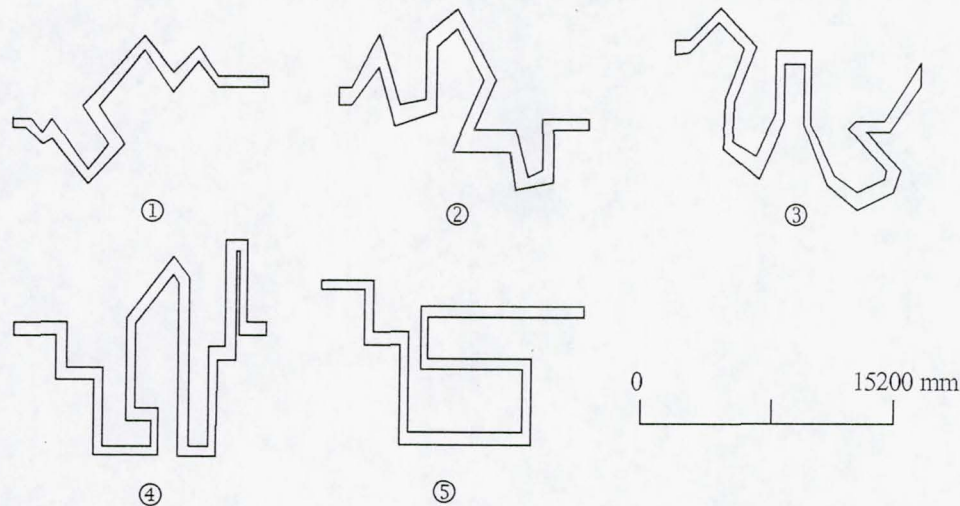


Figure 2 The Five Shapes of Paths Used in Our Experiments

3.4 Interaction Design

The *flying vehicle control metaphor* [17] was used in this work to aid in performing the body-centric navigation tasks. An advantage of using this metaphor is that physical locomotion is not required, so that the user can travel a long distance without leaving the seat. By manipulating a joystick, the subject can travel around the scenario. The mappings for both wayfinding and travel are linear. Considering that it is important to implement constraints and limit degrees-of-freedom (DOF) without reducing significantly the user's comfort, we restricted travel to a fixed level relative to the floor of the room. The head tracker, however, is operated in six DOFs. Additionally, a mismatch of movement along the head direction might occur when the subject looks in other directions. Hence, we force the subject to look forward while traveling and to stop traveling while looking around. In certain situations, the subject can still fly around freely, look in any direction when staying still, and tilt his/her head in any orientation. To perform manipulation and selection, the classical *virtual hand metaphor* [18] was employed in this work. An object can be selected by "touch" using a virtual hand. We mapped the scale and position of the subject's physical hand directly to the scale and position of a virtual hand linearly for hand-centric selection and manipulation.

Numerous papers discuss different interaction techniques in virtual environments, such as "put-that-there" [19], flash light technique [20], World-In-Miniature [21], or Scaled World Grab [22]. They were not applied in our first stage of the design, because we do not want the subject's behavior in the virtual environment to go beyond the human's capability in the physical world. So at this point, the power of VE is used to duplicate the physical world, not to extend the subject's abilities to perform tasks impossible in the real world.

In order to provide effective visual feedback, an information-rich model was built to make the objects *smart*. Objects are *smart* in terms of their response to the subject's interaction. For example, during exposure, the subject needs to know which object can be picked up. Therefore, an object is wire-framed if it can be picked up (only one object is wire-framed each time); highlighted and wire-framed if it has been picked up; and appears to mark the destination platform on which the object should be deposited. For example, Figure 3 illustrates that a wire-framed green ball is picked; it is also rendered with specular highlights. The path appears once the object is picked, so that the subject knows that he must go through this path to put the ball on the platform on which there is an image. If the ball is dropped in range, the wire-frame, highlights, and the image will disappear. Another object will be wire-framed, and this process repeats until the subject finishes the current level or runs out of time.

3.5 Codelets

To make the VE adaptive to all subjects and models, we designed three script files, called *subject codelet*, *application codelet*, and *level codelet*. Figure 4 is an example of a complete subject codelet that defines the subjects' information. The use of this codelet can:

- (1) define the initial locomotion and orientation or viewpoints of the subject in the virtual scenario;
- (2) select the model of user's hand for picking up (depends on the subject is left or right hand dominant); and
- (3) configure the moving scalar in six degree of freedom to decide how fast the subject can move and rotate.

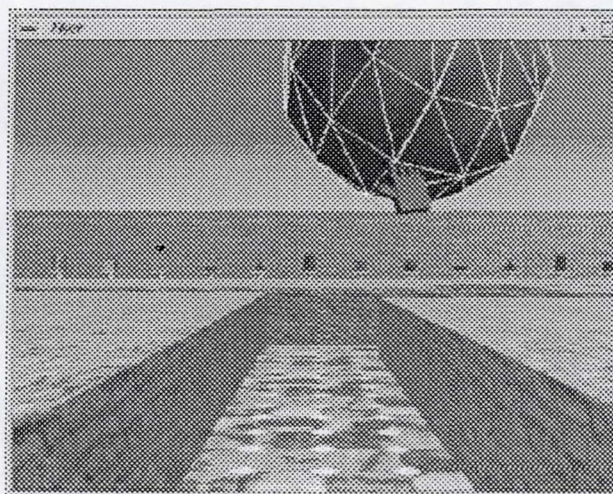


Figure 3 A Screen Shot of the Virtual Scenario (Level 2)
The path and the object are *smart* objects.

*initialpos	100.0	100.0	0.
*view	0.0	0.0	60.0
*model	"hand.ASE" "hand"		
*motion	5	5	5 0 0

Figure 4 Subject Codelet

We allow the operator to load different subject codelet in the application codelet. In this codelet (Figure 5), the operator can:

- (1) choose the filename of the player to be loaded;
- (2) determine which level codelet to be loaded.

Finally, in *level codelet* the operator can:

- (1) choose the appropriate application codelet;
- (2) choose the names and order of the objects to be moved;
- (3) choose the collidable objects; or
- (4) choose the destination platform for each movable object with token '*match'. If this token is omitted, the destination platform will be chosen in random order. An

*subject	"0010.sub"
*level	"level1.lv" "Level"

Figure 5 Application Codelet

example of a level codelet is illustrated in Figure 6. All names in the double quotes are the corresponding object names defined in the model file (level1.ASE in this example).

*model	"level1.ASE"
*ground	"walls"
*movable	"Torus01"
*movable	"Cylinder03"
*collidable	"walls"
*collidable	"Floor"
*collidable	"path01"
*collidable	"path05"
*match	"Torus01" "column01"
*match	"Cylinder03" "column05"

Figure 6 Level Codelet

Each codelet is given different tokens to define different behaviors. Without recompiling the program, the operator can specify the visual databases, define objects' behaviors, define interaction metaphors, and input subject data (e.g., hand-head distance, arm distance, etc.). Table 1 lists all tokens used in our codelets. More detail regarding their usages can be found in Chen [23]. One of the big advantages of the codelet is that there is no need to reprogram the whole system if other visual

Model	Ground	Path
Match	ObjectPath	MatchHighLighObj
Level	Movable	Collidable
Player	Comments	InitialPos

Table 1. Tokens in Codelets

databases are to be loaded or the behaviors of the objects are to be changed. Re-writing a codelet to define the name of the model file and the behaviors (for collision, pickup etc.) will work fine. This should be convenient for instructors or operators who are not programmers.

3.6 Multi-model Environment

Visual feedback is enhanced by auditory feedback to increase realism. In our system, we have implemented sonification, that is, using 2D sound to provide useful information. For example, different sounds are played when the visual database has been loaded or when the subject picks up an object, releases an object correctly or incorrectly, hits an obstacle, is sick, or finishes the trial. Thus, our system is a multi-model system since it integrates visual and auditory feedbacks. The subject is taught to understand the different sound effects in the environment before exposure.

3.7 System Architecture

OpenGL Performer executing on an SGI Onyx® provides the image generation systems. To render correct images on the spherical surface of the dome, we used the Spherical Projection of Image (SPI) Application Program Interface (API) from Elumens [15] to render the distorted images. Since the Onyx does not support auditory output, a PC serves as a sound server. The simulator architecture is illustrated in Figure 7. Instructor/operator represents the person who is in charge of the overall physical and virtual environments during exposure. His/her job is to take the subject through a carefully designed exposure procedure and record data. Since most of them are not programmers, we built an Instructor/Operator Interface (IOI), a graphics user interface (GUI), which provides a virtual platform for the operator. All commands can be issued through IOI by clicking buttons or filling out a form. For example, a *subject information menu* (Figure 8)

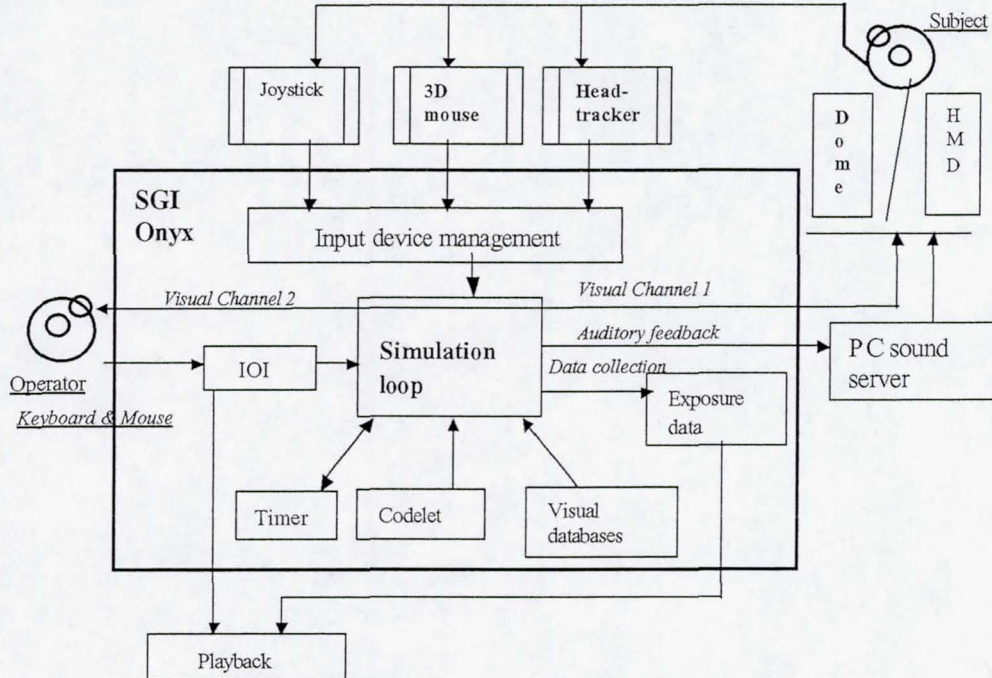


Figure 7 Simulator Architecture

The screenshot shows a graphical user interface window titled "sub info". It is divided into several sections. On the left, there are three sections: "Session Option" with three radio buttons labeled "The First", "The Second", and "The Third"; "Exposure Duration" with three radio buttons labeled "30 minutes", "18 minutes", and "12 minutes"; and "Device Option" with two radio buttons labeled "HMD" and "DOME". On the right, there is a section titled "Printer Info" which contains five text input fields: "Subj. #:" (containing "0200"), "Subj. ID:" (containing "1000"), "AGE:" (containing "25"), "GENDER:" (containing "F"), and "COMMENTS:" (containing "15.10.2011"). At the bottom of the window, there are four buttons: "Create", "Load", "Exit", and "Help".

Figure 8 Subject Information Menu

allows the operator to choose different sessions, exposure durations, and the type of the VE delivery systems as well as to define subject identification and other information.

The *simulation loop* is the kernel of our system. It communicates with the other five modules (except the playback module) when running the simulation. The main Performer function parses several predefined codelets, then renders the scene. The loop repeats a series of actions for the duration of the main function. These actions manage the control of the application in each cycle based on the codelets. The simulation loop also captures events from a number of input devices, and then updates the visual and auditory feedbacks. The operator, who manages the running process, has the right to issue commands through the IOI interface to start or terminate the simulation loop.

During subject exposure, two levels are loaded alternatively. To decide which level is run, and how long the next simulation loop can be run, a *timer* module has been implemented. It controls the switch between level one and level two. Different visual databases and script files are loaded when running different levels. For example, assume the subject is exposed to a 30-minute session. When the program starts, the timer informs the simulation loop to load

the level one database and the corresponding script files. If subsequently the subject finishes this level in 12 minutes or less, the timer module will leave level one, load level two, and most importantly inform the simulation loop how many remain. If the subject finishes level two in less than 18 minutes, the timer restarts the level one loop. Otherwise, it will terminate this session.

Like other virtual environment applications, our system integrates numerous input devices. The *input device management module* provides an interface to the simulation loop. In each cycle, the loop gets the input data from the input devices, e.g., joystick and the trackers. The sensorial output module captures these inputs and generates the corresponding visual and auditory stimuli. The subject experiences coherent feedback according to the instantaneous context. Finally, the data collection module records exposure and performance data in corresponding files.

The *playback module* is independent of the simulation loop. It gets the data from the data collection module and replays the exposure process in both 2D and 3D scenarios. The operator is capable of specifying a subject identification and a session name through the IOI and can replay the exposure process of that subject.

3.8 Data Collection and Playback Tool

Traditional data collection is done using videotaping, which has proven useful in postural stability research, or a written account. Subjective reporting using questionnaires is also a very common technique [24]. However, it is done *after* the trial and we must assume that the subject retains detailed memories of each part of the experience. Unfortunately, this is not always true. Our method complements this technique by recording data *while* the subject is immersed in the virtual environment.

Once the exposure data are collected, we need a way to simulate the exposure process. A playback module implements this function. We present two views: one has been implemented as a 2D graph, and the other is a 3D view. The 2D graph is drawn by *gnuplot* and displays the path the subject flew through during a selected exposure. The 3D view presents both overviews (global view) and a life-size virtual environment (local view) (Figure 9). This replay visualizes the subject's operations during the run. All these operations can be effected through the IOI by clicking buttons.

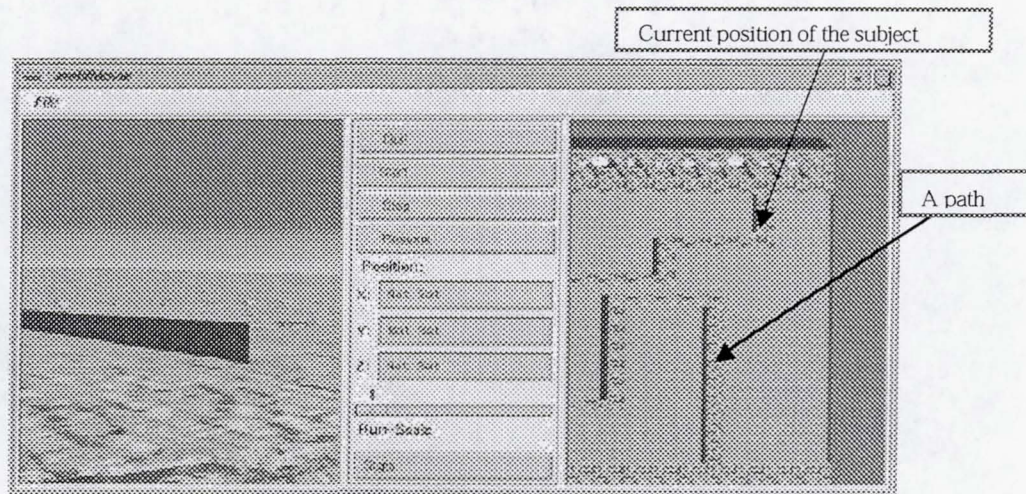


Figure 9 Playback Interface

The data collection module was implemented in software, without interfering with the subject exposure. We simply record every operation, current physical position, and other parameters of interest in an ASCII file. The instructor can open the ASCII file later through IOI for review or playback. This method can be used together with videotape and written documents to investigate subject performance. The data we record include various aspects of the trial, such as task completion time, task accuracy, the subject's current position in each loop, the name of the currently selected object, the name of the designated platform, collision errors, selection errors, and sickness status.

4. Pilot Study Design

This is an on-going project, and the pilot study on using our system will be executed starting in June 2002 and lasting about one year. The final results are expected to be available in July 2003. We have designed the pilot study to determine the extent to which cybersickness occurs and sensorimotor functions are degraded as a function of the type of VE delivery system, repeated exposures, and length of exposure. Three experiments (Table 2) are designed to compare responses to both types of VEs using a mixed (within- and between-subjects) experiment design. Each of the experiments is designed to examine one of the sensorimotor functions: (1)

Experiment	Independent Variables	Dependent Variables	Tests	Equipment
1	Display type; VE order; exposure duration	VOR; eye and head velocity; saccades, etc.	Visual target acquisition; pursuit tracking	EOG Amps; cruciform LED target system
2	Display type; VE order; exposure duration	Manual angular displacement; constant error	Eyes open and eyes closed manual pointing	Proxima™ system; laser pointer
3	Display type; VE order; exposure duration	COB; LOS; DI	Open and closed loop posture tests	Chattecx™ balance system

Table 2. Experiment Summary

eye-head coordination, (2) eye-hand coordination, or (3) postural equilibrium. Experiments 1-3 are designed to examine each of the three sensorimotor responses as a function of length of exposures and repeated exposures.

5. Results and Conclusion

Although major results are still forthcoming, initial responses to the simulator have been very positive. Most users appreciate the setup of the simulator, and describe situations where its features would be useful in the study of different display system. The user-friendly interfaces have also been well received. Figure 10 presents pictures of a pilot study. The left picture demonstrates an HMD test case while the right one demonstrates the Dome test case. In the pilot study, the same physical environment setup was kept for both systems.

One of our purposes when designing the system was to maintain interaction fidelity; therefore we disallowed interaction behaviors that were impossible in the real world. We conclude that the VET is helpful in measuring the subject's performance and susceptibility to cybersickness, and to study the aftereffects of VE exposure. The system has the potential to serve as a tool with which to build additional evaluation experiments. We are among the first who evaluate the display devices.

6. Future Work

During the development of the VE system, we realized the necessity of developing a uniform virtual environment testbed. At this point, it should support all known interaction components and integrate various input and output devices.

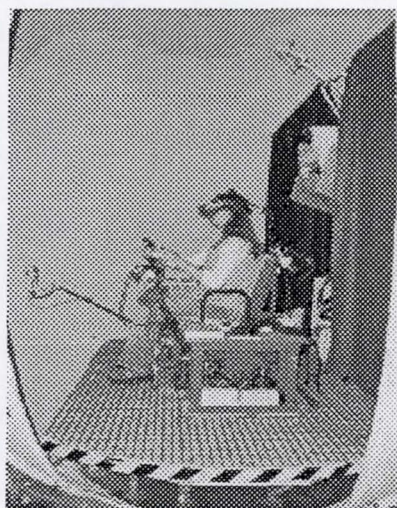


Figure 10 Physical Environment Setup of HMD and Dome

An important area of future work is to conduct a formal user evaluation. One goal of the formal evaluation is to determine the benefits of display devices; another goal is to validate the appropriateness of the interface design.

7. Acknowledgments

This work is supported by NASA NRA -98HEDS-02. We would like to thank Dr. D. Bowman, at Virginia Tech, for his answering of queries; C. Cao for setting up an initial version of the system; R. King and H. Garcia, at Old Dominion University, for their technical support and beautiful model creation; and Dr. H.B. Duh, in NASA/JSC, for his comments.

References:

- [1] Loftin, R.B. and Kenney, P., "Training the Hubble Space Telescope Flight Team," *IEEE Computer Graphics & Applications*, 15(5): 31-37, 1995.
- [2] Chen, J., Fang, Y., Loftin, B.L., Leiss, E.L., Lin, C., and Su, S., "An Immersive Virtual Environment Training System on Real-Time Motion Platform," *Proc. of the Conference on Computer Aided Design & Computer Graphics*, China, 951-954, 2001.
- [3] Vince, J., "Virtual Reality Systems," Addison-Wesley, 1995.
- [4] Sutherland, I.E., "The Ultimate Display," *Proc of the IFIP Congress*, 65(2): 506-508, 1965.
- [5] Brooks, F.P., "What's Real About Virtual Reality," *IEEE Computer Graphics & Applications*, 19(6): 16-27, 1999.
- [6] Bryson, S., Zeltzer, D., Bolas, M.T., Chapelle, B.D., Bennett, D., and Spitz, E.L., "The Future of Virtual Reality: Head Mounted Displays Versus Spatially Immersive Displays," *SIGGRAPH'96 Panel Session*, 1996.
- [7] Van Dam, A. Forsberg, A.S., Laidlaw, D.H., LaViola, J.J., and Simpson, R.M., "Immersive VR for Scientific Visualization: A Progress Report," *IEEE Computer Graphics & Applications*, 20(6): 26-52, 2000.
- [8] Mine, M., "Virtual Environment Interaction Techniques," *Tech. Reports, TR95-018*, UNC, Department of Computer Science, 1995.
- [9] Bowman, D. and Hodges, L.F., "An Evaluation of Techniques for Graphing and Manipulating Remote Objects in Immersive Virtual Environments," *Proc. of the ACM Symposium on Interactive 3D Graphics*, 35-38, 1997.
- [10] Howard, I.P., "Spatial Vision within Egocentric and Exocentric Frames of Reference," *In S.R. Ellis (Ed.), Pictorial Communication in Virtual and Real Environments*, London: Taylor & Francis, 1993.
- [11] Bowman, D., Datey, A., Farooq, U., Ryu, Y.S., and Vasnaik, O., "Empirical Comparisons of Virtual Environment Displays," *Tech. Report TR01-19*, Virginia Polytechnic Institute & State University, Department of Computer Science, VA, 2001.
- [12] Pausch, R., Shackelford, M.A., and Proffitt, D., "A User Study Comparing Head-mounted and Stationary Display," *Proc. of the IEEE Symposium on Research Frontiers in Virtual Reality*, 41-45, 1993.
- [13] Pausch, R., Proffitt, D. and Williams, G., "Quantifying Immersion in Virtual Reality," *Proc. of SIGGRAPH'97*, 13-18.
- [14] Personal Communication Between Bowman, D. and Chen, J. in 2001.
- [15] <http://www.elumens.com>.
- [16] Lampton, D.R., Knerr, B.W., Goldberg, S.L., Bliss, J.P., Moshell, J.M., and Blau, B.S., "The Virtual Environment Performance Assessment Battery (VEPAB): Development and Evaluation," *Presence*, 3(2): 145-157, 1994.
- [17] Ware, C. and Osborne, S., "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments," *Computer Graphics*, 24(2): 175-183, 1990.
- [18] Robinett, W. and Holloway, R., "Implementation of Flying, Scaling, and Grabbing in Virtual Worlds," *Proc. of the ACM Symposium on Interactive 3D graphics*, 189-192, 1992.
- [19] Boleter, R., "Put-that-there": Voice and Gesture at the Graphics Interface," *Proc. of SIGGRAPH'80*, 262-270, 1980.
- [20] Liang, J. and Green, M., "JDCAD: A Highly Interactive 3D Modeling System," *Computer & Graphics*, 4(18): 499-506, 1994.
- [21] Stoakley, R., Conway, M.J., and Pausch, R., "Virtual Reality on a WIM: Interactive Worlds in Miniature," *Proc. of CHI'95*, 265-272, 1995.
- [22] Mine, M.R., Frederick, J., Brooks, P., and Sequin, C.H., "Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction," *Proc. of SIGGRAPH'97*, 1997.
- [23] Chen, J. "A Virtual Environment System for the Comparative Study of Dome and HMD," Master Thesis, Department of Computer Science, University of Houston, 2002.
- [24] Kennedy, R.S., Jones, M.B., Lilienthal, M.G., and Harm, D.L., "Profile Analysis of After Effects Experienced During Exposure to Several Virtual Reality Environments," *AGARD Conference Proc. 541 - Virtual Interfaces: Research and Applications*, 2-1-2-9, 1993.